

```

1 //[城市正視圖/Urban Elevations]
2 #define IN "P0514IN.txt"
3 #define OUT "P0514OUT.txt"
4 //*****
5 #include <iostream>
6 #include <time.h>
7 using namespace std;
8 void redir(void);
9 //*****
10 /* Work Space*/
11 #include <algorithm>
12 struct Building{
13     int id;
14     double x, y, w, d, h; //(x, y), width, depth, height
15
16     bool operator < (Building &rhs) { // *this < rhs
17         return x < rhs.x || (x == rhs.x && y < rhs.y);
18     }
19 }b[100]; //根據題意
20 double x[100*2];
21 int n;
22
23 bool visible(int i, double mx);
24 bool cover(int i, double mx);
25 //*****
26 int main(void)
27 {
28     redir(); //redirection
29 //*****
30 /* Work Space*/
31     int i, j;
32     int m;
33     int kase = 0;
34     bool vis;
35
36     while(scanf("%d", &n) == 1 && n){
37         for(i=0; i<n; i++){
38             b[i].id = i+1;
39             scanf("%lf%lf%lf%lf%lf", &b[i].x, &b[i].y, &b[i].w, &b[i].d, &b[i].h);
40             x[i*2] = b[i].x;
41             x[i*2+1] = b[i].x + b[i].w;
42         }
43         sort(b, b+n); //將建物排序
44
45         sort(x, x+n*2); //劃分出區間
46         m = unique(x, x+n*2) - x; //x座標排序後去除重複，得到m個座標
47
48         if(kase++){
49             printf("\n");
50         }
51         printf("For map #%d, the visible buildings are numbered as follows:\n%d", kase, b[0].id);
52         for(i=1; i<n; i++){
53             vis = false;
54             for(j=0; j<m-1; j++){
55                 if(visible(i, (x[j]+x[j+1])/2)){
56                     vis = true;
57                     break;
58                 }
59             }
60             if(vis){
61                 printf(" %d", b[i].id);
62             }
63         }

```

```

64     printf("\n");
65 }
66 /**
67     freopen("CON", "r", stdin); //取消重新導向
68     freopen("CON", "w", stdout);
69
70     printf("Time used = %.2f\n", (double)clock()/CLK_TCK); //傳回程式目前為止執行的時間
71
72     system("pause");
73     return 0; //the end...
74 }
75
76 void redir(void)
77 {
78     freopen(IN, "r", stdin);
79     freopen(OUT, "w", stdout);
80 }
81 /**
82  /* Work Space*/
83  //判斷建築物i在x=mx處是否可見
84  bool visible(int i, double mx)
85  {
86      int k;
87
88      if(!cover(i, mx)){
89          return false;
90      }
91
92      for(k=0; k<n; k++){
93          if(b[k].y < b[i].y && b[k].h >= b[i].h && cover(k, mx)){
94              return false;
95          }
96      }
97
98      return true;
99 }
100
101 //判斷建築物i是否包含這個mx區間
102 bool cover(int i, double mx)
103 {
104     return b[i].x <= mx && mx <= (b[i].x+b[i].w);
105 }
106
107 //Input(IN) Sample
108 /*
109 14
110 160 0 30 60 30
111 125 0 32 28 60
112 95 0 27 28 40
113 70 35 19 55 90
114 0 0 60 35 80
115 0 40 29 20 60
116 35 40 25 45 80
117 0 67 25 20 50
118 0 92 90 20 80
119 95 38 55 12 50
120 95 60 60 13 30
121 95 80 45 25 50
122 165 65 15 15 25
123 165 85 10 15 35
124 0
125 */

```